**Hardware Inventory**

A Project Report Submitted to

**Concordia University Wisconsin**

In partial fulfillment of the requirement for the award of the degree of

**MASTER OF SCIENCE**

**IN**

**INFORMATION TECHNOLOGY**

**BY**

[AAMER SALEEM MOHAMMED](mailto:aamersaleem.mohammed@cuw.edu) (F00445450)

**TO**

[PROFESSOR JACOB HOPPE](mailto:jacob.hoppe@cuw.edu)(Client)

UNDER THE GUIDANCE OF

**Professor Jacob Hoppe**

(Visiting Professor of CS and Assistant Director of IT dept.)



**DEPARTMENTOF INFORMATION TECHNOLOGY**

**CONCORDIA UNIVERSITY WISCONSIN (**NCA- accredited)

* A Lutheran higher education community

IT Integrative Capstone (MS-IT CSC 565)

Table of Contents

[Project Abstract 4](#_Toc479759298)

[System analysis 5](#_Toc479759299)

[Existing System: 5](#_Toc479759300)

[Disadvantages of existing system: 6](#_Toc479759301)

[Proposed System: 6](#_Toc479759302)

[Advantages of proposed system: 6](#_Toc479759303)

[Literature Survey 7](#_Toc479759304)

[Feasibility Study 7](#_Toc479759305)

[Economic feasibility: 8](#_Toc479759306)

[Technical feasibility: 8](#_Toc479759307)

[Social feasibility: 8](#_Toc479759308)

[System Requirements 9](#_Toc479759309)

[Application Requirements 9](#_Toc479759310)

[Functional Requirements and Requirement Ranking 9](#_Toc479759311)

[Hardware Requirements 10](#_Toc479759312)

[Software Requirement 11](#_Toc479759313)

[User Interface Requirements 11](#_Toc479759314)

[Other Requirements 11](#_Toc479759315)

[Project and product constraint 11](#_Toc479759316)

[Process Timeline 12](#_Toc479759317)

[Q&A Transcript 12](#_Toc479759318)

[Client 12](#_Toc479759319)

[Third-party 14](#_Toc479759320)

[Anticipated steps and difficulties 18](#_Toc479759321)

[System Specifications 18](#_Toc479759322)

[Environment 18](#_Toc479759323)

[Front-end development: 19](#_Toc479759324)

[Back-end development: 19](#_Toc479759325)

[User Interface 19](#_Toc479759326)

[Wireframe Prototype (Using Bootstrap) 20](#_Toc479759327)

[System Functions Defined 21](#_Toc479759328)

[Network Scan 21](#_Toc479759329)

[Display Device List 22](#_Toc479759330)

[Quick Search 22](#_Toc479759331)

[Filter Options 22](#_Toc479759332)

[Export to Excel 22](#_Toc479759333)

[Supporting Tools, Libraries and API 22](#_Toc479759334)

[Bootstrap 22](#_Toc479759335)

[C# API 22](#_Toc479759336)

[C# SNMP Library 22](#_Toc479759337)

[Acceptance Criteria 22](#_Toc479759338)

[System Design 23](#_Toc479759339)

[Visual representation of prototype components: 23](#_Toc479759340)

[Pseudocode 24](#_Toc479759341)

[Component Diagram 26](#_Toc479759342)

[System Architecture 26](#_Toc479759343)

[Data Flow Diagram 26](#_Toc479759344)

[Use Case Diagram 27](#_Toc479759345)

[System Implementation 29](#_Toc479759346)

[User module: 29](#_Toc479759347)

[Screens divisions: 30](#_Toc479759348)

[Server Module: 30](#_Toc479759349)

[Files and Resources: 30](#_Toc479759350)

[Implementing code from Visual studio on Windows Server 2008 R2 (Virtual Machine) 31](#_Toc479759351)

[Sample code: 35](#_Toc479759352)

[Webpage source code (Front End) 35](#_Toc479759353)

[Network Scanning Module 42](#_Toc479759354)

[Back-End 42](#_Toc479759355)

[Simple Network Management Protocol (SNMP) 49](#_Toc479759356)

[Bootstrap source code: 50](#_Toc479759357)

[Java Script: 50](#_Toc479759358)

[CSS: 50](#_Toc479759359)

[Fonts (Icons): 50](#_Toc479759360)

[Dyna Table source code: 50](#_Toc479759361)

[Performance and usability improvement 51](#_Toc479759362)

[System Testing 51](#_Toc479759363)

[Types of tests: 51](#_Toc479759364)

[Unit testing 51](#_Toc479759365)

[Integration testing 52](#_Toc479759366)

[Functional test 53](#_Toc479759367)

[White Box Testing 53](#_Toc479759368)

[Black Box Testing 54](#_Toc479759369)

[Acceptance Testing 54](#_Toc479759370)

[Conclusion: 55](#_Toc479759371)

[References 56](#_Toc479759372)

# Project Abstract

The Computer Science department of Concordia University Wisconsin is a proudly growing Christian faith environment. This growth has made us improvise and add many tools and techniques for us to grow in mind, body and spirit. Hopefully this improvement will continue further.

To support the structure of the tools we hold in CS department, specifically the machines and devices connected to the network of CS department, there is yet no management tool to keep track and information on these devices. Therefore, here lies a problem. Being in professional IT organization, it is very important to keep track of all these devices. Managing this information in a presentable and updatable read-only manner is the purpose of this project. This purpose would solve the problem of not being able to know about the basic yet informative data of devices such as Device type, Device ID, Device OS, Device IP, Device Model, Manufacturer etc., additional problems such as not knowing whether a device is connected to network, being able to know about certain machine without physically trying to go the device are some of the problems that will be solved.

# System analysis

System analysis is the analysis of the existing system and future system to get to know attributes for comparing the differences in weight the of advantages and disadvantages between old and new system.

## Existing System:

The existing system being used is very time consuming and unreliable when compared to the proposed system. There is no formal track record of hardware devices in existing system rather there is use of old school style book and paper which are used to log down every detail of the device to keep them on tab and track the devices in the Computer science department.

### Disadvantages of existing system:

* No remote access.
* Information can be ambiguous at times if not organized properly.
* Insecure with ability to be overwritten.
* No assigned personnel for the task as it is not logical to invest productive time where physical record are kept that can be purged.
* Records can be exposed with no security if not handled properly.
* No historical data storage ability.
* Of course, no automatic update of data when devices are added.
* Records can only be made by a technical and authorized personnel if the records being made require administrative access.

## Proposed System:

The proposed system is a web based system which is accessed through web browser of any computer system of the computer science department. The server, devices and the machine on which user extracts the information from should be connected to the network mesh. It provides lots of improvements over the existing system, some of the improvements that are requested by the client are mentioned below.

### Advantages of proposed system:

* Remote Access
* Presentable data which is easy to understand
* Detailed information of the devices connected to network
* Automatically shows new devices connected when scanned through network
* Usage of existing hardware devices
* Information cannot be tampered because it is read-only
* Records can be stored in the form of excel files with ease
* All disadvantages of existing system are eliminated in proposed system

# Literature Survey

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool they will need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is considered for developing the proposed system.

## Feasibility Study

The feasibility of the project is analyzed in this phase and project proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to operate. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* Economic feasibility
* Technical feasibility
* Social feasibility

### Economic feasibility:

This study is carried out to check the economic impact that the system will have on the organization. Thus, the developed system is well within the budget with only very few and easily available general hardware and software resources being used and this was achieved because most of the technologies used are freely available.

### Technical feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement; as only minimal or null changes are required for implementing this system.

### Social feasibility:

The aspect of this study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

# System Requirements

## Application Requirements

### Functional Requirements and Requirement Ranking

The following are details of functional requirements for the application

|  |  |
| --- | --- |
| **FR1** | Scan Network Devices |
| **Input** | -- |
| **Output** | List of Devices in network with device information |
| **Function** | Scan all devices in the local network and for each device query device information |
| **Priority** | HIGH |

|  |  |
| --- | --- |
| **FR2** | Display Device List |
| **Input** | List of scanned devices |
| **Output** | Tabular data of list of all scanned/filtered devices |
| **Function** | Display list of scanned devices in tabular form. |
| **Priority** | HIGH |

|  |  |
| --- | --- |
| **FR3** | Quick Search |
| **Input** | Search keyword, scanned device list |
| **Output** | List of Devices with name or IP address matching search keyword |
| **Function** | Filter device list based on search keyword matching device name or IP address |
| **Priority** | HIGH |

|  |  |
| --- | --- |
| **FR4** | Filter Option |
| **Input** | Filter keys for device type and/or manufacturer, scanned device list |
| **Output** | List of Devices matching filter keys for device type and/or manufacturer |
| **Function** | Filter device list based on selected device type and/or manufacturer |
| **Priority** | HIGH |

|  |  |
| --- | --- |
| **FR5** | Export to Excel |
| **Input** | Tabular data of scanned devices (all or filtered) |
| **Output** | Excel file containing data of scanned devices |
| **Function** | Export the tabular data into excel file |
| **Priority** | MEDIUM |

|  |  |
| --- | --- |
| **FR6** | Reports |
| **Input** | -- |
| **Output** | Reports shows some statistical information |
| **Function** | Display reports like pie chart showing total devices based on device types etc. |
| **Priority** | MEDIUM |

### Hardware Requirements

* A minimal configuration server with appropriate operating system, processing power and RAM will be enough as a host for the application.
* As this application needs to scan network devices, the server on which the application will be deployed should have network access and all necessary ports open to query network devices.
* User will need a computer machine on the same network to access the application over the network.

### Software Requirement

* As the application is web based, the server should contain a web server (ex: IIS) to host the application.
* Server should contain necessary libraries, framework and runtime environment (.NET 4.0) related to technology used to developed the application.
* Server should contain necessary libraries used to query network devices like SNMP library.
* As the application is web-based, user needs a web browser to access the application.

### User Interface Requirements

* Simple and effective UI which can be used by both technical and non-technical users.
* Information of the discovered devices to be displayed in a logical and organized manner.
* Discoverable devices are to be shown in a Clear and Organized manner with as much as Effective and Informative information which needs to be responsive to a non-technical user as well as a technical user.

### Other Requirements

* As this project is web based, it might be possible to provide access through internet, but not necessarily.
* User requires the previous data to be wiped out when new data is extracted.
* User needs read-only data.
* Secure access to the product is not necessary.

## Project and product constraint

Hardware Inventory is a very popular product required and used by organizations to keep track of their technological inventory. It is used very vastly in various areas of IT industry and IT support. Other areas like schools, colleges, hospitals also require this product. Due to its popular usage, there are only few constraints to this product as per the market’s general requirement. Miscellaneous basic constraints are, constant network connection, constant running of host machine and getting familiar of the project to a non-technical person as understanding this project is complex for a non-technical person.

Other product constraints are agent availability on the devices that needs to be scanned, required access privileges to the machines (both server and other network devices) and there might be no connection to devices that are powered off or are disconnected from the network. The biggest constraint would be how to display the extracted data of the connected devices in an efficient, understandable and proper organized manner.

## Process Timeline

Requirements(output) and problem analysis: 12 Feb 2017 - 13 Mar 2017

System needs and requirements: 1 Mar 2017 - 20 Mar 2017

Designing, prototyping and developing: 10Mar 2017 - 27\*Mar 2017

Documenting: 12 Feb 2017 - 31 Mar 2017

Implementation and Testing: 25 Mar 2017 - 4 April2017

Evaluation and Maintenance: 4 April 2017 - Deadline

## Q&A Transcript

### Client

1. What is the problem? Why is there a problem?

**Response**: No inventory system on server for CS department.

Current inventory system which is possibly paper based is becoming inconvenient due to growing inventory.

1. Why do you Need it?

**Response:** To know what devices are connected to internet and what are their basic details.

1. Why Remote access?

**Response**: To have ease of access to helping to decrease the physical strain and activity and provide extended range of access.

1. Purpose of hardware inventory?

**Response**: To provide information when with presentable content to know and keep track of network connected devices.

1. What format does the output be required to organize and display?

**Response**: Textual information in a presentation format is required.

1. Previous attempts?

**Response**: NONE.

1. What expectations with UI?

**Response**: Clear & to the point, Familiar & Consistent, Effective and informative

1. What type of devices in a network would you like to discover?

**Response**: All of them.

1. Where to deploy the product?

**Response**: On CS department server running IIS Windows server.

1. What kind of database would you prefer to be used to store extracted data?

**Response**: Any.

1. Where is the initial deployment to be done?

**Response**: Initial prototype deployment is to be done on private computer then after success with that a Virtual server machine will be provided to work on.

1. Do we have access to all the devices we need to operate?

**Response**: No, permission required.

1. Should the queries be automated?

**Response**: No. should run on demand.

1. How technical should the information should be?

**Response**: Not too technical but also not too basic.

### Third-party

#### Meeting outline with Professor Spencer

The purpose of meeting with prof. spencer was to understand the expectation and perspective of a third-person who in not involved in the project when they are asked to let us know what do you expect from this project. The focus of the meeting would be on user Interface, functions and features of the project. Understanding the suggestions of a third person and evaluating their response to the proposal planned by the developers to make the project better are some of the perks achieved through this meeting.

#### First Meet on 8 Mar, 2017

The meeting was initiated with making the third person understand what the project was about and what we were told to achieve; that is demonstrating the project succinctly and describing the project abstract.

##### Q & A:

1. What to you understand when you hear hardware inventory?

**Response**: Speaking as a technical person, I would think that you will be checking the devices connected to the server via any computer connected to that server and show them the required details of those devices in a way that is easy to understand and informative.

1. What do you expect from hardware inventory?

**Response:** Speaking hypothetically as your client, I would expect every single detail of the devices connected to the network and that is speaking as a technical person.

1. Why do you think client needs hardware inventory?

**Reason:** For necessity to know what we have connected in computer science department and to keep track of what we have as there are many devices.

1. What problems do you think the clients proposed requirements will solve?

**Response:** This project would solve the problem of keeping less reliable physical record and provide digital records with better management functionality and would provide a platform to track hardware devices saving lot of time and cost.

1. How are you expecting to interact with hardware inventory?

**Response:** I think this would be a web based application so I expect to see the results of UI on a webpage.

1. How do you expect to access hardware inventory?

**Response:** I would presume that I will be able to have remote access to the application through which I can access web application maybe within the university or on the devices directly connected to the server.

1. What do you not want in hardware inventory?

**Response:** Complex UI should be avoided and UI should be user friendly for even a non-tech savvy person to understand.

1. What specific features you want in it?

**Response:** Being able to know details of the devices such as Vendor, IP address and MAC address might be the primary attributes that user would want to see.

1. What specific features would you like to have?

**Response:** Helpful tips of what each function exactly does if the function is too complex would be a good add-on.

1. Who will be using or expected to be using it?

**Response:** People with authority and higher level access would be expected to use it.

1. Do users of this application have technical knowledge?

**Response:** I would like to think that they would have minimal technical knowledge.

1. As a technical person, where do you think we will face more problems?

**Response:** Dilemma of what technology that you will be working on will be a challenge till deployment phase of the project, as there are wide ranged supportive technologies that would support various functionality of the user requirements. Getting access privileges will also be a challenge, checking if every machine is compatible with the technology that the project is made on will face some bumps on estimated project timeline.

1. As a technical person, how would you like us to approach these issues?

**Response:** First try and finalize the technology that you will be working on as choosing the technology will take up some time and then make stable timeline referring to the steps of SDLC.

1. How frequently do you think you would need it?

**Response:** The client would probably need to use this application occasionally with peak usage during semester start.

1. How do you suggest us to manage this project?

**Response:** After knowing the requirement, make a list of requirement and set dates to achieve them so that you are not backed off or extended on one requirement exposing the whole project to danger.

1. How do you think we should plan our work on documentation?

**Response:** Working in a cloud environment for documentation would be my suggestion that you can always keep track and share whatever the work is going on and be up to date on your achievements. Always follow the scheduled documentation with progress of the project.

#### Second meet on Mar 27, 2017

Our second meet with professor spencer was a virtual meeting via skype where we demonstrated our UI and discussed the technologies that we were implementing. Demonstrating the UI and functionality of the project prototype and recording and evaluating the response from professor was the main purpose of this meet for us.

##### Q & A:

1. Is this UI a comfortable way to satisfying client’s requirement?

**Response:** Yes, web browser is a popular tool on every computer with an essence to familiarity to their user, so I think UI representation is comfortable.

1. What are your views on web page designed?

**Response:** The web page designed is simple and informative with spacious and elaborative view with few and well-arranged GUI and user interactive elements.

1. Do you have any suggestions for improving the web-page design?

**Response:** No, all the content on the presently designed web-page is fine. Just double-check with your client though.

1. Do you see any design issue of the prototype demonstrated?

**Response:** I see that the project is still incomplete and need to done as soon as possible being stumbled with the time constraint and decide the other module of technology, other than that the UI developed is good per the requirements for the project.

1. Do you think the technology that this project is being implemented on is appropriate?

**Response:** Speculating the functionality and support with the OS, the technology is very likely to be suitable for implementation on IIS server.

## Anticipated steps and difficulties

* Select development technology that will be used to develop the application. Technology in which open source and free libraries are available will be our preference while selecting the technology.
* Find best suited SNMP library in the selected technology which will be used to collect information from the network devices.
* Develop the module related to scan local network for devices and for each device fetch required information.
* Develop user interface and integrate with the module to implement the required features.

# System Specifications

## Environment

Users using this tool will be mostly network administrators who have knowledge about networking as well as knowledge about CS network. This will help users in identifying the device within the CS network.

In addition to above, our application must also be used by a novice user who doesn’t have much knowledge about networking but is familiar with basic hardware terms.

Our aim will be to keep the design in such a way that even a novice user can get some useful information from it. Therefore, a user can deploy this on any network with .NET framework and check hardware inventory.

As our application will be web based, therefore user need to have knowledge about how to deploy it in their environment.

Environment need for this application is a computer with decent configuration connected to local network and a web server on that computer to deploy. Thereafter, only an internet browser is required to access this application anywhere in the network.

The Software requirements that we are finalizing to fulfill the client requirements are divided into two parts:

* Front-end web development
* Back-end development

### Front-end development:

For front end web development or client-side development we will be producing webpages using HTML, CSS and JavaScript. For doing this, we will be specifically using an open source framework called “Bootstrap” to help us create the webpages needed for this project. (Bootstrap, n.d.)

### Back-end development:

For back end development, we will be using a Virtual Machine provided by client which will be running on windows server (IIS). The application that is to run this machine will be created with the help of “Visual Studio” using .NET framework. There will be several other supporting Dynamic-link libraries, C# based open source SNMP library for .NET to be exact, will be used including some Application Programming Interface (API) using C#; all these resources being used will be Open Sourced and community based which require no licensing. (Khan, n.d.) (lextm, n.d.) (Visual Studio Downloads, n.d.)

## User Interface

The primary output of this application is a tabular data showing list of hardware devices on the network. This application also requires some input in the form of text and events through input elements like textbox, button etc.

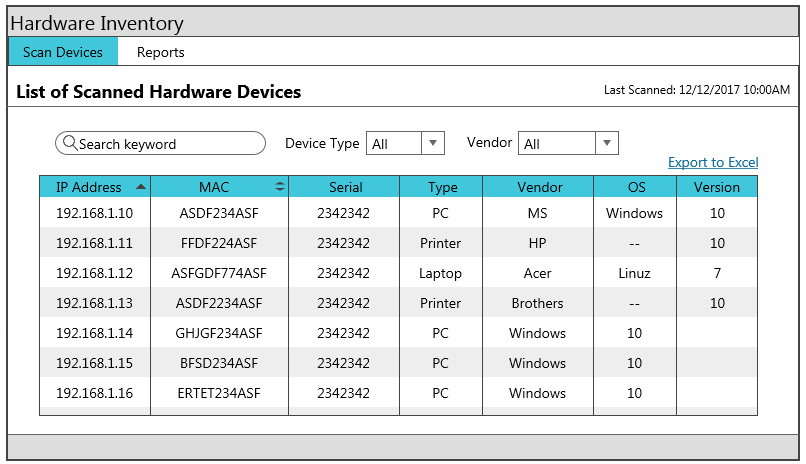
Now a day's most of the users are familiar with web based user interface, be it on computer or mobile. This application being a web-based system, will not be different. It will use simple UI containing standard elements like text box, button, dropdown list etc.

The application flow will be through web pages and hyperlinks where user can make transition between pages by clicking on appropriate hyperlinks.

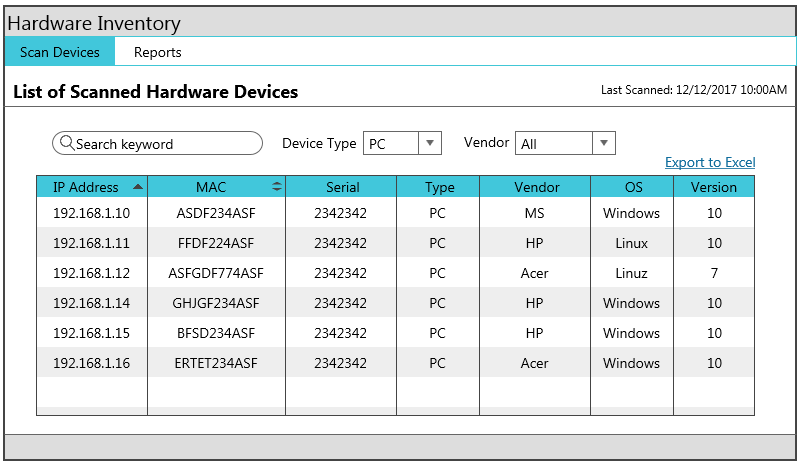
Application layout will contain header and footer which will be same throughout the application. Content area will change according to user selection. A menu will be provided into content area for transition between pages/features. These required webpages with their functions will be produced using Bootstrap.

### Wireframe Prototype (Using Bootstrap)

The following screenshots shows wireframe of the user interface than can be expected:



Above screenshot shows list of all scanned devices with search and filter options. It also sq1111111111111hows last scanned date and time and an option to export the result in excel format. There is a search text box in with user can enter device name or IP address to search.



Above screenshot shows an example where user filter scanned devices by device type. In the screenshot user, has selected PC as the device type and the list of scanned devices only shows PCs.

## System Functions Defined

Following is the list of system functions:

### Network Scan

Quickly & easily scan for hardware devices within local area network. Application provide two types of scanning option:

#### Automatic Scan

One is automatic scan which is done whenever the application is started.

#### Manual Scan

Another one is manual refresh in which user can rescan the network devices by clicking refresh button.

Last scanned date and time will be displayed on top right corner to keep the user informed when the last network scan was performed.

### Display Device List

Display list of scanned devices with basic information like device name, IP address, manufacturer, device type, serial number etc.

### Quick Search

Quick search facility will allow user to quickly search for a device by device name or IP address.

### Filter Options

Easy filter options to view devices by manufacturer and by device type. By default, all devices will be displayed. When a specific filter option is selected, the application will show only those devices.

### Export to Excel

Export list of scanned devices (all or filtered) devices in excel format.

## Supporting Tools, Libraries and API

### Bootstrap

Boot strap is a popular front-end development tool that we will be using in our project because of its vast community support, open source licensing and its faster and easier web development libraries and templates. It’s easy and efficient single code base, based on HTML, JavaScript and CSS will make working with it a good assistance. (Bootstrap, n.d.)

### C# API

Searching through the vast databases and communities, an API that resembles the project requirement was found on which the project can be worked on as a referring point and then be collaborated with our custom requirement and molded to our code. This code is based on C#. (Khan, n.d.)

### C# SNMP Library

The Library is designed to be used in all kinds of applications such as Web Forms (including .NET Compact Framework), ASP.NET, and WPF\* running on Microsoft .NET and Novell Mono platforms not including Silverlight as it lacks a lot of socket support that #SNMP relies on. It can be separated first into two parts, Messaging and MIB. The Messaging part focuses on SNMP messages while the MIB part focuses on SNMP MIB documents. This library will provide us most of the required essential functions that are to be satisfied.

## Acceptance Criteria

The primary feature of this application is to scan & list hardware devices within the local network. If the application is successful in doing so, then the other features are assistive features where it helps the users in viewing the information effectively. If we accomplish this feature into the application, then it will be a major part completed.

# System Design

## Visual representation of prototype components:

Server

Send and Receive data Send and Receive data

Remote computer (View Data)

Local Computer

(View Data)

Send and Receive data

Other network devices

Printer

Local computer (View Data)

This visual representation is a high-level abstract of currently existing connections forming a network in the current situation. The data being sent and received are high level terms for queries and responses between these devices. The queries are being requests from the client machine which can be any of the above local computers and the other machines in the network being at the pinging end of the data.

## Pseudocode

1. **START**
2. **CHECK**

**IF** the network is running **THEN**

Goto step 3

**ELSE**

Exit

1. **CHECK**

**IF** all the devices have proper framework and agent **THEN**

Goto step 4

**ELSE**

Exit

1. **CHECK**

**IF** the client machine is host machine **THEN**

Goto step 5

**ELSE**

Establish connection with host machine

1. **OPEN** Web browser
2. **IF** client is not on host machine **THEN**

**ENTER** host IP and get UI through web browser

**ELSE**

**OPEN** web page directly and get UI through web browser

1. UI component **interacts** with the **Web app**
2. ASP. NET based web app **interacts** with **network library** in IIS
3. **INITIAL AUTOMATIC SCAN** lists all the devices in a tabular format
4. A server-side back-end execution runs parallel with device scan to scan printers.
5. **CLICK** on refresh button to rescan the network and update devices data
6. **FILTER** device via search box with any device information column
7. **SORT** devices per Name, Type or IP
8. **EXPORT** data on an excel sheet
9. **END**

## Component Diagram

### System Architecture

Server

User Machine

IIS (Web server)

Network Library

Web Browser

HTTP

ASP .NET Webapp

AJAX

Scan Network Devices

Printer

Printer

Computer

Computer

Computer

## Data Flow Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

Local computer or Server

Accessing via web browser

IF server

True False

Server

Functional Web page

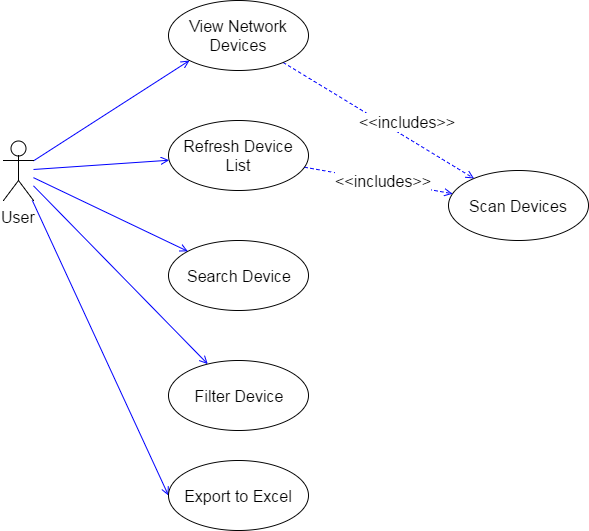
Quick Search

Device List(default)

Filter

Sort

## Use Case Diagram



# System Implementation

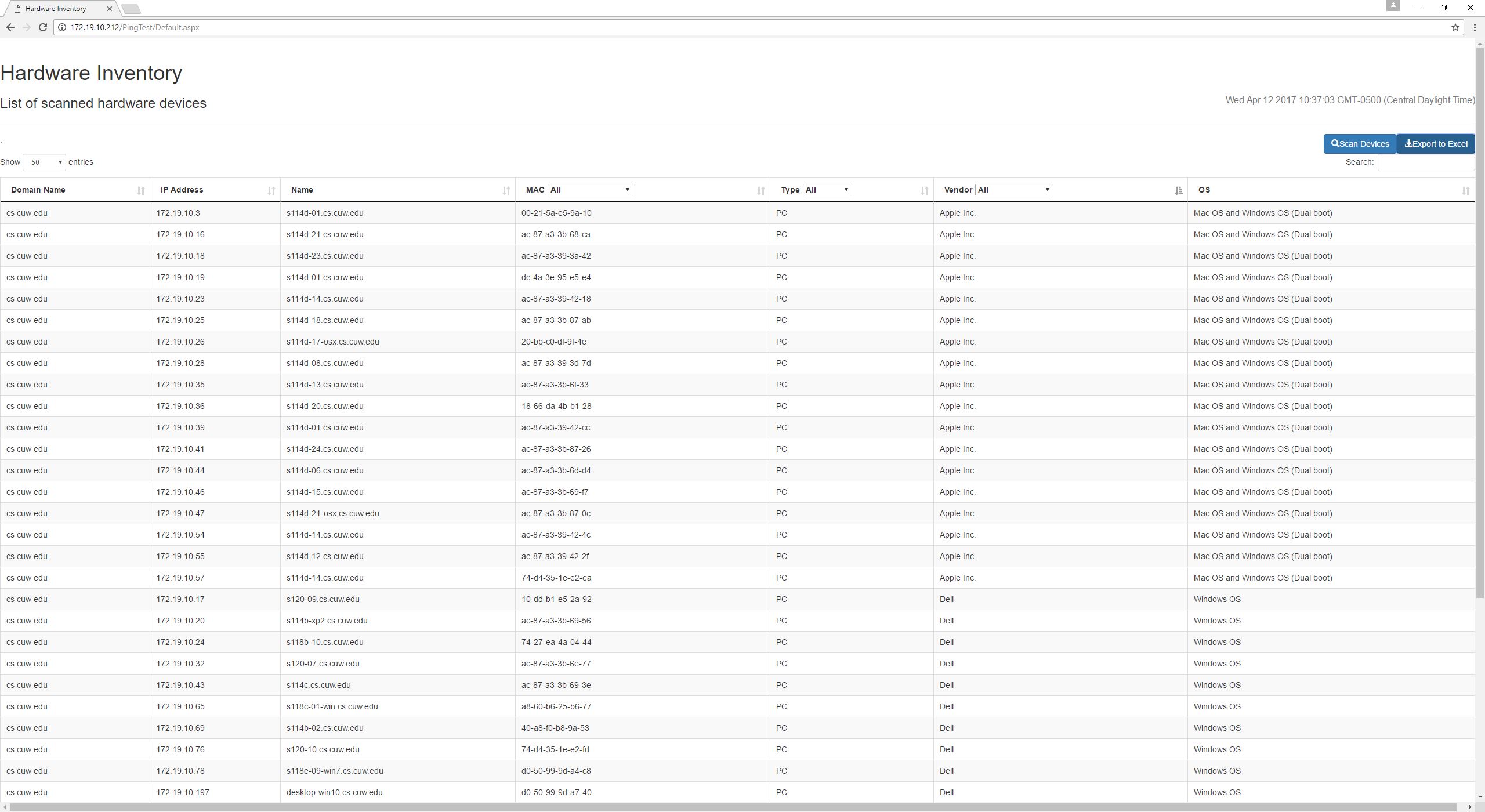
Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

## User module:

In this module, users should can search details and browse through the information about hardware devices. Before searching and accessing these details, user should at least have minimum knowledge of how to explore the and understand the system.

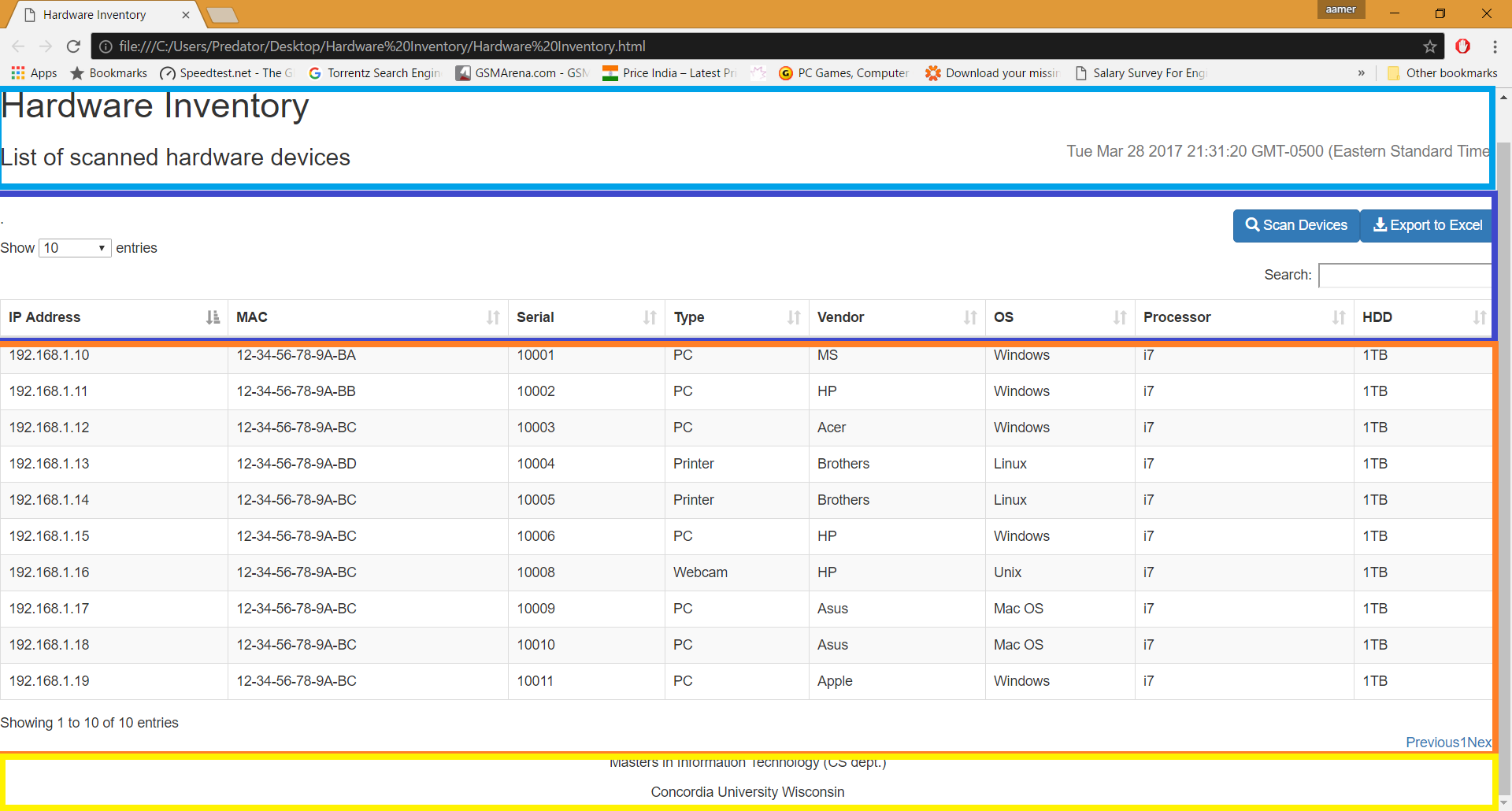
The user is taken to only one webpage which is easy to understand with basic functionality of filtering data, sorting data and exporting data in the form of an excel file. Other passive informational functions are showing current time with time zone.



### Screens divisions:

The screen is divided into four sections:

* The top section of the webpage shows the header with time.
* The second section is the functional section where all the user interactive elements are present.
* The third section is the main section which contains data of the devices in the form of a table with ability to navigate through data elements if they exceed the amount of shown elements.
* The final section of the webpage is the footer.



## Server Module:

### Files and Resources:

Resources are the files that are used to build up the front-end webpage and the back-end source code development. There are two categories of resources:

#### Internal resources:

The internal resources are files copied to in the storage of the local server. These files will stream down from the server and get locally cached at the client side. If the player loses internet connectivity, the resource will still be usable with no interruption in access and downtime of the web application.

#### External resources:

The external resources are files residing at external repositories, like a web server. You can link to these external files and use the files to get the load off server. Loading resources from an external web server decreases the server side processing and instead the processing is done on the client side.

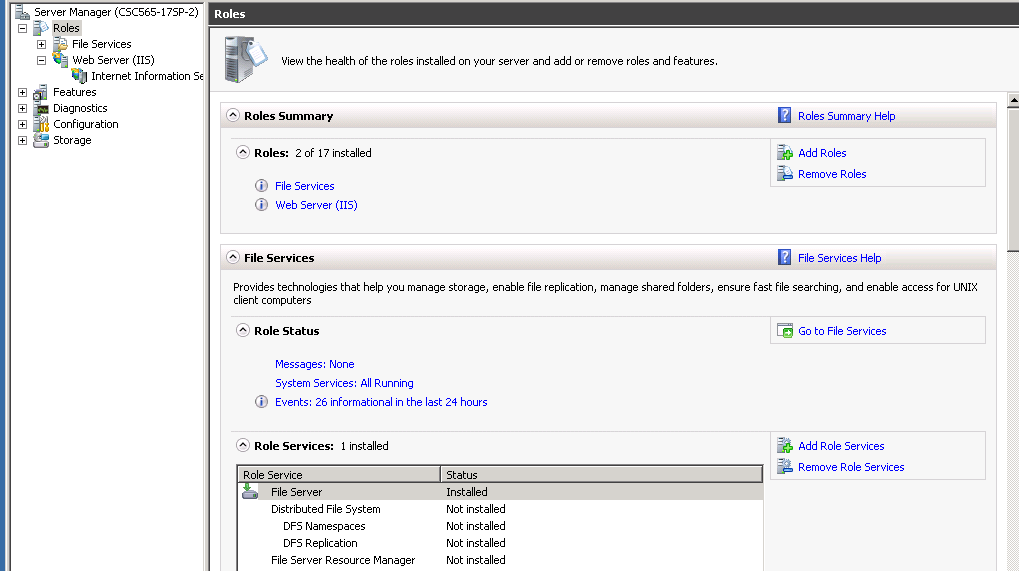
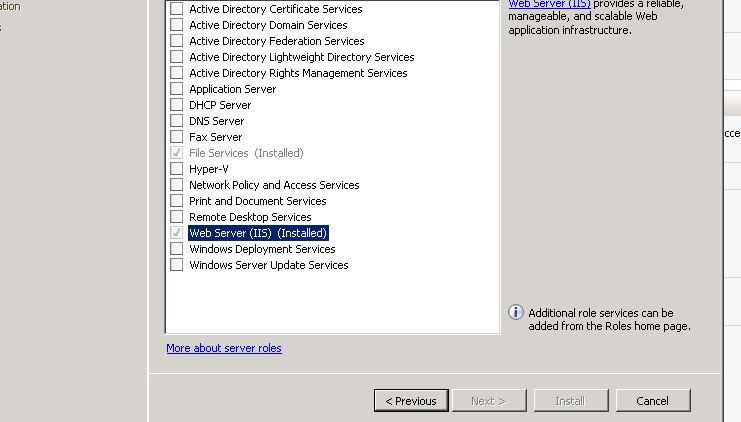
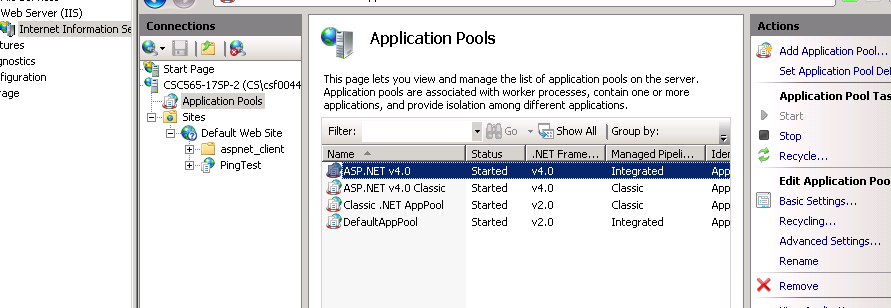
The main difference between the external and internal files is the internal resources take advantage of our caching system; which means; whenever these resources are executed they are cached in local server for faster access after their initial access. If the connection to the server is interrupted, the internal resources will continue and run on the local server.  
  
The external resources are not downloaded onto the server; therefore, they are not cached. However, in some circumstances it is easier to work and swap external resources. This also increases flexibility of the web application.

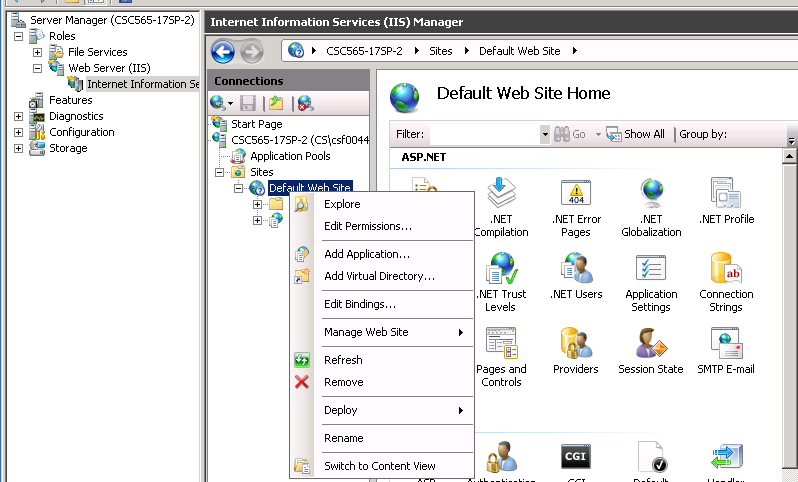
To conclude there are trade-offs, of pros and cons equally good features with no extensive influence on the usage and interactive capability of the user.

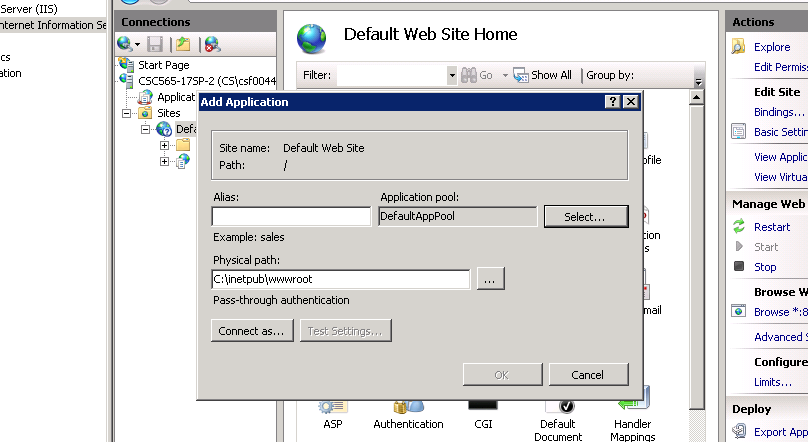
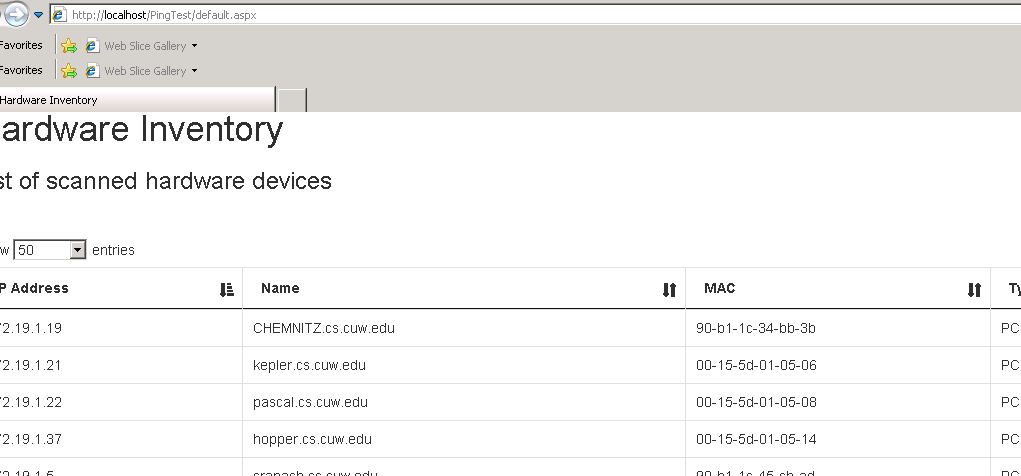
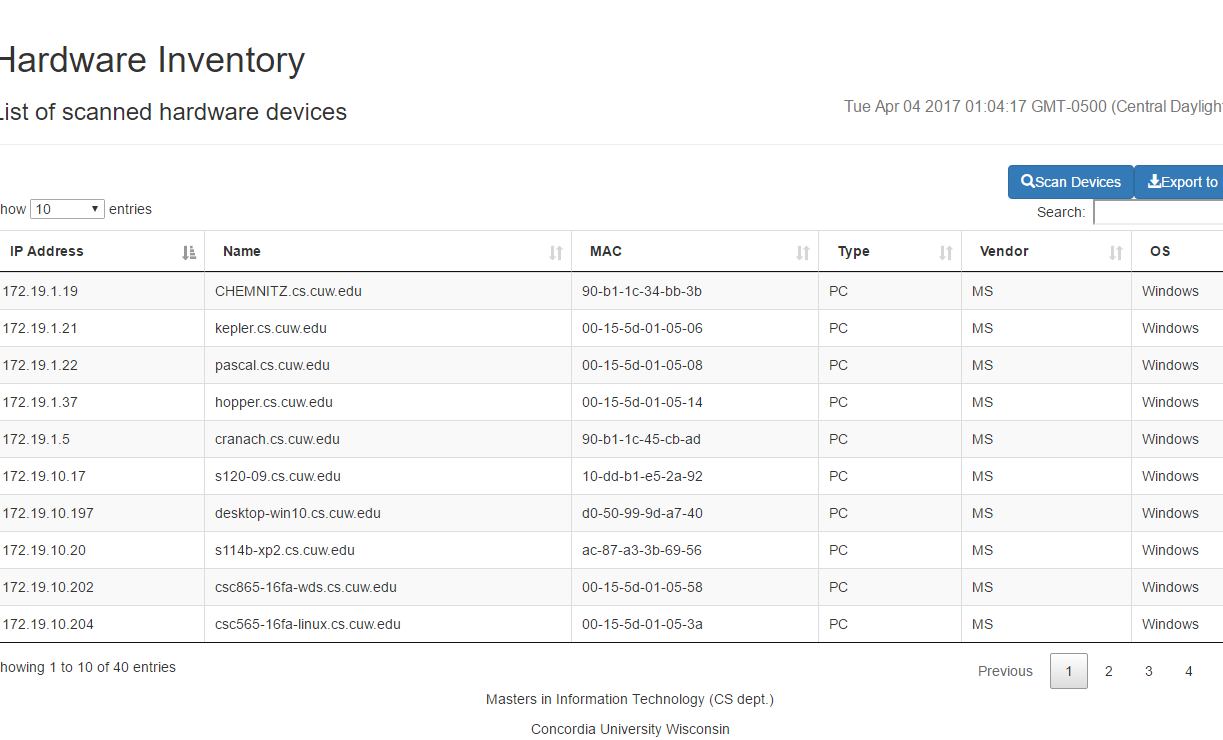
The devices that are scanned are stored in. JSON file extension which are then processed and displayed in a table formal on a HTML webpage.

Other supporting resources are supporting files and libraries of Bootstrap, Dyna Table and C# SNMP library for .NET. Other components include AJAX and WMI libraries.

## Implementing code from Visual studio on Windows Server 2008 R2 (Virtual Machine)

1. Copy the project file from visual studio “Projects” folder.
2. Open IIS
3. Open Server Manager
4. Check in the left tab under “Roles” if there is “Web Services (IIS)” present
5. If it is not there, click on Add Roles on the right-hand side and Add Web server (IIS).
6. After web services installed expand Web services option under Roles and click on Application Pools under Connection to check if the ASP .NET 4.0 Framework is available. If not go to this [link](https://www.microsoft.com/en-us/download/details.aspx?id=17851) and install it and restart the server for changes to take effect an then you should be able to see ASP .NET listed in application pools. (Microsoft .NET Framework 4 (Web Installer), n.d.)
7. Right click on the “Sites” folder under Connections tab and select “Add Application”.



1. Under Physical path select “C:\inetpub\wwroot” and select ASP .NET 4.0 under Application pool and click ok. This will be the path where the web application files are copied to.
2. Copy and paste the web application source folder under “C:\inetpub\wwroot”
3. Open Web browser on the server and in address bar type <Http://localhost/Websitename(PingTest)/deefault.aspx> and press Enter. 
4. The application will load and done.
5. This web application this query all the devices and display them on the screen from any computer. To operate this application on other computers connected to servers, Open web browser and type <http://172.19.10.212/PingTest/Default.aspx> in the address bar.

## Sample code:

### Webpage source code (Front End)

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="PingTest.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-type" content="text/html; charset=utf-8">

<meta name="viewport" content="width=device-width,initial-scale=1">

<title>Hardware Inventory</title>

<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">

<link rel="stylesheet" type="text/css" href="css/bootstrap.dataTable.min.css">

<link href="css/jquery.dataTables.min.css" rel="stylesheet" />

<style type="text/css" class="init">

</style>

<!--Adding java source files-->

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/jquery.dataTables.min.js"></script>

<script src="js/dataTables.bootstrap.min.js"></script>

<script src="js/jquery.table2excel.js"></script>

</head>

<body>

<form id="form1" runat="server">

<div>

<header class="page-header">

<!--Header-->

<h1>Hardware Inventory</h1>

<h3>List of scanned hardware devices<span><small><p id="datetime" style="float: right;"></p>

</small></span></h3>

</header>

<!--Date and Time-->

<script>document.getElementById("datetime").innerHTML = Date();</script>

</div>

<div>

<%-- <select id="os\_select" class="form-control pull-left">

<option value="">All OS</option>

<option>Windows</option>

<option>Linux</option>

<option>Mac OS</option>

</select>--%>

<!--Scan and Export buttons-->

<div class="text text-center" id="loading">

<span class="alert alert-info" style="padding:8px;"><img src="img/hourglass.gif" style="padding:0px 5px;width:32px;" />Scanning devices (Found: <span id="foundCount">0</span>)</span>

</div>

<button type="button" class="btn btn-primary pull-right" id="btn-export"><span class="glyphicon glyphicon-download-alt"></span>Export to Excel </button>

<button type="submit" class="btn btn-primary pull-right" id=""><span class="glyphicon glyphicon-search"></span>Scan Devices </button>

</div>

<p>.</p>

<!--Table-->

<div class="fw-container">

<div class="fw-body">

<div class="content">

<table id="example" class="table table-striped table-bordered" cellspacing="0" width="100%">

<thead>

<tr>

<th>Domain Name</th>

<th>IP Address</th>

<th>Name</th>

<th>MAC</th>

<th>Type</th>

<th>Vendor</th>

<th>OS</th>

</tr>

</thead>

<tbody id="exampleBody">

</tbody>

<tfoot>

<tr>

<th>Domain Name</th>

<th>IP Address</th>

<th>Name</th>

<th>MAC</th>

<th>Type</th>

<th>Vendor</th>

<th>OS</th>

</tr>

</tfoot>

</table>

</div>

</div>

</div>

<asp:HiddenField ID="HiddenField1" runat="server" />

</form>

<footer>

<center> <!--Footer-->

<p>Masters in Information Technology (CS dept.)</p>

<p>Concordia University Wisconsin</p></center>

</footer>

<!--Table to Excel and Data table script-->

<script>

$(document)

.ajaxStart(function () {

$('#loading').show();

})

.ajaxStop(function () {

$('#loading').hide();

});

var table;

$(document).ready(function () {

$('#exampleBody').html($('#<%=HiddenField1.ClientID%>').val());

$('#<%=HiddenField1.ClientID%>').val('');

init\_datatable();

snmp\_get("172.19.10.", "1.3.6.1.2.1.1.1.0");

$('#btn-export').on('click', function () {

$('<table>').append(table.$('tr').clone()).table2excel({

exclude: ".excludeThisClass",

name: "Worksheet Name",

filename: "Report" //do not include extension

});

});

})

function init\_datatable() {

table = $('#example').DataTable({

"initComplete": function (settings, json) {

this.api().columns([4, 5, 6]).every(function () {

var column = this;

var heading = $(column.footer()).text();

var select = $('<select style="margin:0px 5px;"><option value="">All</option></select>')

.appendTo($(column.header()).empty().text(heading))

.on('change', function () {

var val = $.fn.dataTable.util.escapeRegex(

$(this).val()

);

column

.search(val ? '^' + val + '$' : '', true, false)

.draw();

});

column.data().unique().sort().each(function (d, j) {

select.append('<option value="' + d + '">' + d + '</option>');

});

});

}

});

}

function snmp\_get(ip\_subnet, oid) {

for (i = 0; i <= 255; i++) {

var ip = ip\_subnet + i;

$.ajax({

type: "GET",

url: "SnmpGet.ashx?oid=" + oid + "&ip=" + ip,

success: function (data) {

if (data != "") {

$('#foundCount').text(parseInt($('#foundCount').text())+1);

processData(ip, data);

}

}

});

}

}

function processData(ip, data1) {

var oid = "1.3.6.1.4.1.11.2.3.9.1.1.7.0";

var hostName = "NA";

var mac = "NA";

var type = "NA";

var vendor = "NA";

var os = "NA";

console.log("processing data1:" + data1);

var data1Array = data1.split('#');

console.log("using ip:" + data1Array[0]);

if (data1.indexOf('HP') >= 0) {

var url = "SnmpGet.ashx?oid=" + oid + "&ip=" + data1Array[0];

console.log('url:' + url);

$.ajax({

type: "GET",

url: url,

success: function (data) {

console.log("processing data:" + data);

//var jsonData = data;// JSON.parse(data);

var dataArray = data.split('#');

var ip1 = dataArray[0];

if (dataArray[1] != "") {

var attrArray = dataArray[1].split(';');

jQuery.each(attrArray, function (index, item) {

var itemArray = item.split(':');

console.log("key:" + itemArray[0] + ";val:" + itemArray[1]);

if (jQuery.trim(itemArray[0]) == 'MFG') {

vendor = itemArray[1];

}

if (jQuery.trim(itemArray[0]) == 'DES') {

hostName = itemArray[1];

}

if (jQuery.trim(itemArray[0]) == 'CLS') {

type = itemArray[1];

}

});

//var tr = $("<tr><td>" + ip + "</td><td>" + hostName + "</td><td>" + mac + "</td><td>" + type + "</td><td>" + vendor + "</td><td>" + os + "</td></tr>");

//$('#exampleBody').append(tr);

//init\_datatable();

//table.row.add({

// "IP Address": ip,

// "Name": hostName,

// "MAC": mac,

// "Type": type,

// "Vendor": vendor,

// "OS": os

//}).draw(false);

ip1 == "172.19.10.69" ? mac = "40:a8:f0:b8:9a:53" : "";

ip1 == "172.19.10.3" ? mac ="00:21:5a:e5:9a:10":"";

table.row.add(["cs cuw edu",

ip1,

hostName,

mac,

type,

vendor,

os

]);

//table.init();

table.destroy();

init\_datatable();

}

}

});

}

}

</script>

</body>

</html>

### Network Scanning Module

There are some options available to us which we can use to implement the network scanning module in ASP.NET C#.

#### ASP.NET System.Net package

Microsoft ASP.NET provides a package System.Net which provides lot of functions to perform network related operations like

* Get network interfaces of the computer
* Ping a computer
* Get computer name
* Get computer mac address

### Back-End

using Lextm.SharpSnmpLib.Messaging;

using SnmpSharpNet;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Management;

using System.Net;

using System.Net.NetworkInformation;

using System.Net.Sockets;

using System.Threading;

using System.Web;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

namespace PingTest

{

public partial class Default : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

// exampleBody.InnerHtml = "";

String ServerIP = GetServerIP();

String GatewayIP = GetNetworkGatewayIP();

BulkPing(ServerIP);

if (GetSubnet(ServerIP) != GetSubnet(ServerIP))

{

BulkPing(GatewayIP);

}

//Label1.Text+=GetOS("172.20.11.179");

}

public string GetSubnet(String IP)

{

String[] array = IP.Split('.');

return array[2];

}

public static string GetServerIP()

{

IPHostEntry ipHostInfo = Dns.GetHostEntry(Dns.GetHostName());

foreach (IPAddress address in ipHostInfo.AddressList)

{

if (address.AddressFamily == AddressFamily.InterNetwork)

return address.ToString();

}

return string.Empty;

}

static string GetNetworkGatewayIP()

{

string ip = null;

foreach (NetworkInterface f in NetworkInterface.GetAllNetworkInterfaces())

{

if (f.OperationalStatus == OperationalStatus.Up)

{

foreach (GatewayIPAddressInformation d in f.GetIPProperties().GatewayAddresses)

{

ip = d.Address.ToString();

}

}

}

return ip;

}

public void BulkPing(String IP)

{

string[] array = IP.Split('.');

for (int i = 2; i <= 255; i++)

{

string HOST\_IP = array[0] + "." + array[1] + "." + array[2] + "." + i;

PingHOST(HOST\_IP, 1, 4000);

}

}

public void PingHOST(string HOST\_IP, int count, int timeout)

{

//Label1.Text += "ping:" + host + "<br/>";

for (int i = 0; i < count; i++)

{

new Thread(delegate()

{

try

{

System.Net.NetworkInformation.Ping ping = new System.Net.NetworkInformation.Ping();

ping.PingCompleted += new PingCompletedEventHandler(PingCompleted);

ping.SendAsync(HOST\_IP, timeout, HOST\_IP);

}

catch(Exception ex)

{

// handle exception

}

}).Start();

}

}

private void PingCompleted(object sender, PingCompletedEventArgs e)

{

string IP = (string)e.UserState;

if (e.Reply != null && e.Reply.Status == IPStatus.Success)

{

string DomainName = GetDomainName(IP);

string HostName = GetHostName(IP);

string MACAddress = GetMacAddress(IP);

string[] arr = new string[3];

string Vendor = HostName != null ? (HostName.IndexOf("565") > 0 ? "Virtual Machine" : (HostName.IndexOf("s114d") == 0 ? "Apple Inc." : "Dell")) : "";

string OS = HostName != null ? (HostName.IndexOf("s114d") == 0 ? "Mac OS and Windows OS (Dual boot)" : (HostName.IndexOf("osx.") > 0 ? "Mac OS and Windows (Dual boot)" : (HostName.IndexOf("linux.") > 0 ? "Linux" : "Windows OS"))) : "";

// add new row

//HtmlTableRow row = new HtmlTableRow();

//row.Cells.Add(new HtmlTableCell(IP));

//row.Cells.Add(new HtmlTableCell(HostName));

//row.Cells.Add(new HtmlTableCell(MACAddress));

//row.Cells.Add(new HtmlTableCell("PC"));

//row.Cells.Add(new HtmlTableCell(Vendor));

//row.Cells.Add(new HtmlTableCell(OS));

//example.Rows.Add(row);

HiddenField1.Value += @"<tr>

<td>" + DomainName + @"</td>

<td>" + IP + @"</td>

<td>" + HostName + @"</td>

<td>" + MACAddress + @"</td>

<td>PC</td>

<td>" + Vendor + @"</td>

<td>" + OS + @"</td>

</tr>";

}

else

{

// MessageBox.Show(e.Reply.Status.ToString());

}

}

public string GetDomainName(string ipAddress)

{

try

{

IPHostEntry hostEntry = Dns.GetHostEntry(ipAddress);

String hostName = hostEntry.HostName;

//String ipAddress = hostEntry.

System.Text.StringBuilder sb = new System.Text.StringBuilder();

if (hostEntry != null)

{

char[] separating = { '.' };

String realhostname1 = hostName;

String[] s1 = new string[5];

int i = 0;

bool contains = false;

string[] words = realhostname1.Split(separating);

foreach (string s in words)

{

if (contains = words.Contains("cs"))

{

s1[i] = s;

i++;

}

}

String s2;

for (int j = 1; j < 5; j++)

{

sb.AppendLine(s1[j]);

}

s2 = sb.ToString();

return s2;

}

}

catch (SocketException ex)

{

//Label1.Text += ex.Message;

// MessageBox.Show(e.Message.ToString());

}

return null;

}

public string GetHostName(string ipAddress)

{

try

{

IPHostEntry hostEntry = Dns.GetHostEntry(ipAddress);

String hostName = hostEntry.HostName;

//String ipAddress = hostEntry.

if (hostEntry != null)

{

return hostEntry.HostName;

}

}

catch (SocketException ex)

{

//Label1.Text += ex.Message;

// MessageBox.Show(e.Message.ToString());

}

return null;

}

//Get MAC address

public string GetMacAddress(string ipAddress)

{

string macAddress = string.Empty;

System.Diagnostics.Process Process = new System.Diagnostics.Process();

Process.StartInfo.FileName = "arp";

Process.StartInfo.Arguments = "-a " + ipAddress;

Process.StartInfo.UseShellExecute = false;

Process.StartInfo.RedirectStandardOutput = true;

Process.StartInfo.CreateNoWindow = true;

Process.Start();

string strOutput = Process.StandardOutput.ReadToEnd();

string[] substrings = strOutput.Split('-');

if (substrings.Length >= 8)

{

macAddress = substrings[3].Substring(Math.Max(0, substrings[3].Length - 2))

+ "-" + substrings[4] + "-" + substrings[5] + "-" + substrings[6]

+ "-" + substrings[7] + "-"

+ substrings[8].Substring(0, 2);

return macAddress;

}

else

{

return "OWN Machine";

}

}

public String GetOS(String IP)

{

ManagementNamedValueCollection mContext = new ManagementNamedValueCollection();

mContext.Add("\_\_ProviderArchitecture", 64);

mContext.Add("\_\_RequiredArchitecture", false);

ConnectionOptions options = new ConnectionOptions();

options.Impersonation = ImpersonationLevel.Impersonate;

options.EnablePrivileges = false;

options.Context = mContext;

string strIP = IP;

string sOperatingSystem ="";

string sComputerName ="";

//try

{

ManagementScope ManagementScope1 = new ManagementScope(string.Format("\\\\{0}\\root\\cimv2", strIP), options);

ManagementScope1.Connect();

ObjectGetOptions objectGetOptions = new ObjectGetOptions();

// try

{

ManagementPath mpWin32\_OperatingSystem = new ManagementPath("Win32\_OperatingSystem");

ManagementClass mcWin32\_OperatingSystem = new ManagementClass(ManagementScope1, mpWin32\_OperatingSystem, objectGetOptions);

foreach (ManagementObject moWin32\_OperatingSystem in mcWin32\_OperatingSystem.GetInstances())

{

sOperatingSystem = moWin32\_OperatingSystem["Caption"].ToString();

sComputerName = moWin32\_OperatingSystem["csname"].ToString();

}

}

//catch (Exception ex)

{

// Label1.Text = ex.ToString();

}

}

//catch(Exception ex)

{

// Label1.Text = ex.ToString();

}

return sOperatingSystem;

}

}

}

ASP.Net networking package has some basic operations to perform and have limitation that it can only get basic information and that to only from PC. It cannot ping other devices like printers, hubs etc.

## Simple Network Management Protocol (SNMP)

SNMP another option which can be used to gather information about network infrastructure. It is used to collect information about network devices like computers, printers, hubs etc. It can also be used to manage devices remotely over the network.

SNMP is implemented on the application layer of the network stack. There are many version of SNMP like 1, 2 and 3. Many devices implement SNMP. There are two main components of SNMP i.e. SNMP agent and SNMP manager.

SNMP manager is the host computer which sends queries to other devices for information which will be responded by SNMP agent. SNMP agents are also responsible in storing and managing the data in form of Management Information Base (MIB).

## Bootstrap source code:

Source file for bootstrap can be directly found at <http://getbootstrap.com/getting-started/#download> under the download section with open source licensing.

### Java Script:

Various java script files are used for different functionalities. Further refinement is needed in these files. The files currently being used for operations on webpage and with data on the server side and processing side is extracted from these source files, ”dataTables.bootstrap.min.js”, “jquery-1.7.2.min.js”, “json-records.js”, “jquery.min.js”, “jquery.dataTables.min.js”, “bootstrap.min.js” and “jquery.table2excel.js”.

### CSS:

Style sheets used on the webpage are directly derived from the bootstrap base source code of .CSS files. The .CSS files included are “bootstrap.dataTable.min.css”.

### Fonts (Icons):

Icons used on the Scan devices button and Export to excel button are “Glyphicons” used from Fonts source folder of bootstrap.

(Getting started, n.d.)

### Dyna Table source code:

Dyna table is a library with extensive table functionalities and features through which sorting, searching and organizing data are done in the webpage. Source code for this library can be found at <https://www.dynatable.com> with open source licensing.

#### Java Script:

“jquery.dynatable.js” convert the .JSON dynamic objects from AJAX and shows it in the form of an HTML table

#### CSS:

“jquery.dynatable.css” is the table .CSS file from Dyna table source code which provides visual presentation to the table.

(Dyna Table, n.d.)

#### AJAX

AJAX stands for Asynchronous JavaScript and XML which a feature of sending request to web server and getting response using XmlHttpRequest object in JavaScript. The difference here is that browser doesn't have to wait for the response and it can carry out other tasks therefore allowing user to use the web page even when other parts of the page is getting updated. This feature has revolutionized the use of internet applications.

## Performance and usability improvement

By using AJAX in our application, we can improve performance and usability. As scanning network is a heavy task and it will take lot of time for all devices getting scanned and browser load device list only after the scanning is complete. Using AJAX can divide this big task into small tasks and browser can fire them independently of other. Whenever a response come back from an AJAX request, the list will be updated accordingly. It will appear to the user the list is updated spontaneously and he/she doesn't have to wait for a long time before start seeing the devices.

There are lot of libraries which can be used for this purpose and jQuery is one of the most popular one.

# System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## Types of tests:

### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

#### Features to be tested:

* Manual (refresh button click) scan local network and gather information about all active devices.
* List scanned devices showing basic information like device name, device type, Mac, IP address etc.
* View details of an individual device like vendor, make, model, serial number etc.
* Sort and filter options based on device type, vendor etc.
* Open search option to search device by name, IP address, Mac address etc.
* Export to excel feature to export list of scanned device in excel format (option).

### Integration testing

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

### Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

* Input
* Output
* Functions and
* Interfacing with the system

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

For extra user convenience, drop down menu to some columns in the table were added to sort the data according to selecting made without using search box. These are the columns that will be sorted or searched more often than any other column in the table.

### White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

For white box testing, developers interfered with the dataTables.js file from the java script source code to make a drop down menu with sort functionality. Other intense testing experiments were made to test out the functions that are device specific, but due to permission and lack of privileges set on the virtual machine, String Manipulation of the host name was used to display the device specific details.

The hardest part in white box testing was to show and manipulate code for printers. The problem that was faced during testing was the query was not efficiently parsing the JSON file and querying the IP address of the computer science department printers. It was handled by making the query run on front-end code using ajax without it effecting the performance of the webpage.

### Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Network testing was made to be sure that the web application package deployed on the server is available on the other computer through its URL. The URL to access the web application interface is <http://172.19.10.212/PingTest/Default.aspx>. As one of the client’s requirement was to provide access to this webpage through the campus, it was made sure that it was indeed accessible on every computer on campus by randomly selecting computers from various campus location and making sure the web page is accessible through it.

### Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. To make sure that the project is on the right path, few appointments were made with the client to discuss the user requirements and make sure every aspect of the user interface suffices the user requirement as clients liking. Several changes were suggested by the client during these meet. Other meetings worth a third-party person were also arranged to make sure a user with no knowledge of the project was able to understand what the project does? And how easy it is to interact and gain information from it. Collaborating these responses made the acceptance criteria eased out.

# Conclusion:

In this project, we have proposed a web application for showing information required by the client of the Computer Science Department through a web browser using a rather simple UI webpage.

The purpose of this project was to let client view and store information of the devices that are connected to the CS server, and this project satisfies that with a very versatile a flexible programming language C Sharp while being developed on .NET framework, this assures no issues with compatibility, maintenance or cost because the code is Open sourced, libraries are open source and this is a one-time code application. Based on these factors, we can finally say that the project was a success.

# References

*Application Pool Defaults <applicationPoolDefaults>*. (n.d.). Retrieved from Microsoft: https://www.iis.net/configreference/system.applicationhost/applicationpools/applicationpooldefaults

*Bootstrap*. (n.d.). Retrieved from bootstrap: http://getbootstrap.com/

*Bootstrap 3 Tutorial*. (n.d.). Retrieved from W3Schools: https://www.w3schools.com/bootstrap/

*C# Based Open Source SNMP Library for .NET and Mono*. (n.d.). Retrieved from codeplex: https://sharpsnmplib.codeplex.com/

*Chrome Browser*. (n.d.). Retrieved from Google: https://www.google.com/chrome/

*Default.aspx*. (n.d.). Retrieved from VM IIS server: 172.19.10.212

*Download Notepad++ 7.3.3*. (n.d.). Retrieved from Notepad plus plus: https://notepad-plus-plus.org/download/v7.3.3.html

*Dyna Table*. (n.d.). Retrieved from Dyna Table: https://www.dynatable.com/

*Getting started*. (n.d.). Retrieved from Bootstrap: http://getbootstrap.com/getting-started/#download

*How do I add an application to Default Web Site in IIS 7?* (n.d.). Retrieved from StackOverflow: http://stackoverflow.com/questions/8597123/how-do-i-add-an-application-to-default-web-site-in-iis-7

*https://docs.microsoft.com/en-us/azure/virtual-machines/windows/hero-role*. (n.d.). Retrieved from Microsoft: Experiment with installing a role on your Windows VM

*JavaScript*. (n.d.). Retrieved from Bootstrap: http://getbootstrap.com/javascript/

Khan, J. (n.d.). *How to List all devices info on your WLAN /router Programmatically in C#*. Retrieved from codeproject: https://www.codeproject.com/Tips/889483/How-to-List-all-devices-info-on-your-WLAN-router

*Learn About ASP.NET MVC*. (n.d.). Retrieved from Microsoft: https://www.asp.net/mvc

*Learn About ASP.NET Web Forms*. (n.d.). Retrieved from Microsoft: https://www.asp.net/web-forms

lextm. (n.d.). *C# Based Open Source SNMP Library for .NET and Mono*. Retrieved from CodePlex: https://sharpsnmplib.codeplex.com/

*Microsoft .NET Framework 4 (Web Installer)*. (n.d.). Retrieved from Microsoft: https://www.microsoft.com/en-us/download/details.aspx?id=17851

*The 'Default Web Site' cannot be found on the local computer*. (n.d.). Retrieved from Microsoft: https://technet.microsoft.com/en-us/library/aa997698(v=exchg.80).aspx

*Visual Studio Community*. (n.d.). Retrieved from Microsoft: https://www.visualstudio.com/vs/community/

*Visual Studio Downloads*. (n.d.). Retrieved from Microsoft Visual Studio: https://www.visualstudio.com/downloads/

*Walkthrough: Creating a Local IIS Web Site in Visual Studio*. (n.d.). Retrieved from Microsoft: https://msdn.microsoft.com/en-us/library/a1zz9df4.aspx